```
clear all;
%%%% %%%%%%%%%%%%% Authors-cum-Engineers %%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

MWONGELA D MATHINA: F19/1707/2013

WANJALA N KOTOCHAI: F19/1717/2013

```
%%%%photocoordinates%%%%%%
Model=[210.47 896.96 174.54
       219.92 507.34 195.46
       229.82 206.32 217.02
       578.42 849.63 174.46
       587.52 546.88 188.91
       594.13 243.06 206.49];
X=Model(:,1);
Y=Model(:,2);
Z=Model(:,3);

%%%%%%%%%Ground coordinates%%%%%
G=[670296.32 223343.72  1243.65
    670542.31 223345.03 1259.22
    670745.89 223619.54 1267.65];
Xc=G(:,1);
Yc=G(:,2);
Zc=G(:,3);

%%%%%%%%%%%%%%%% initial values aproximations %%%%%%%%
%%%%%%% obtaining scale %%%%%
            %%%% Ground Magnitude %%%%%

  Grd=(sqrt(((G(3,2)-G(1,2))^2)+((G(3,1)-G(1,1))^2)))*1000;
  Mdl=sqrt(((Y(6,1)-Y(1,1))^2)+((X(6,1)-X(1,1))^2));

    Sc=Grd/Mdl;
 %Computing Kappa
 Md=atan((X(6,1)-X(1,1))/(Y(6,1)-Y(1,1)));
 Gr=atan((Xc(3,1)-Xc(1,1))/(Yc(3,1)-Yc(1,1)));
 K=Gr-Md;
  om=0;
  phi=0;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% customised column matrices to help im matrix formations %%%%%
 M=[1;1;1;2;2;2;6;6;6];      %%%%%% column matrix helps pick photo data whose grou
 T=[3;3;3;4;4;4;5;5;5];      %%%%%% column matrix helps pick photo coordinates who
 C=[1;1;1;2;2;2;3;3;3];      %%%%%% column matrix picks ground coordinates %%%%%%%
 R=[1;1;1;2.5;2.5;2.5;4;4;4];%%%%%% column matrix places the values at their respe

  %%%%%%% Approxmation of the translational elements %%%%%%%
  Model2p=Model(1,:);
```

```
  Model2=[Model2p(1,1);Model2p(1,2);Model2p(1,3)];
  ModelSc=(Sc*Model2)/1000;
  Gp=G(1,:);
  Gt=[Gp(1,1);Gp(1,2);Gp(1,3)];
  Tels=Gt-ModelSc;
  Tx=Tels(1,1);
  Ty=Tels(2,1);
  Tz=Tels(3,1);
  Rt=[Tx;Ty;Tz];

%%%%%%%%%%%% Functions of the three rotations omega,phi and kappa %%%%%

 r11=cos(phi)*cos(K);
 r12=((cos(om)*sin(K))+(sin(om)*sin(phi)*cos(K)));
 r13=(sin(om)*sin(K))-(cos(om)*sin(phi)*cos(K));
 r21=-cos(phi)*sin(K);
 r22=(cos(om)*cos(K))-(sin(om)*sin(phi)*sin(K));
 r23=(sin(om)*cos(K))+(cos(om)*sin(phi)*sin(K));
 r31=sin(phi);
 r32=-sin(om)*cos(phi);
 r33=cos(om)*cos(phi);

for i=1:1

 for i=1:9;
     m=M(i);
     g=R(i);
     i=2*g-1;

%%%%% omega differentials %%%%
A(i,1)=(-Sc*(((-sin(om)*sin(K)+cos(om)*sin(phi)*cos(K))*Y(m))+(cos(om)*sin(K)+sin(
A(i+1,1)=(-Sc*((-sin(om)*cos(K)-cos(om)*sin(phi)*sin(K)*Y(m))+(cos(om)*cos(K)-sin(
A(i+2,1)=(-Sc*((-cos(om)*cos(phi)*Y(m))-(sin(om)*cos(phi))*Z(m)))/1000;

%%%%% phi differentials %%%%
A(i,2)=(-Sc*((-sin(phi)*cos(K)*X(m))+(sin(om)*cos(phi)*cos(K)*Y(m))-(cos(om)*cos(p
A(i+1,2)=(-Sc*((sin(phi)*sin(K)*X(m))+(-sin(om)*cos(phi)*sin(K)*Y(m))+(cos(om)*cos
A(i+2,2)=(-Sc*((cos(phi)*X(m))+(sin(om)*sin(phi)*Y(m))-(cos(om)*sin(phi))*Z(m)))/1

%%%%% kappa differentials %%%%
A(i,3)=(-Sc*((-cos(phi)*sin(K)*X(m))+((cos(om)*cos(K)-sin(om)*sin(phi)*sin(K))*Y(m
A(i+1,3)=(-Sc*((-cos(phi)*cos(K)*X(m))+((-cos(om)*sin(K)-sin(om)*sin(phi)*cos(K))*
A(i+2,3)=0;

%%%%% lambda differentials %%%%

A(i,4)=-((r11*X(m))+(r12*Y(m))+(r13*Z(m)))/1000;
A(i+1,4)=-((r21*X(m))+(r22*Y(m))+(r23*Z(m)))/1000;
A(i+2,4)=-((r31*X(m))+(r32*Y(m))+(r33*Z(m)))/1000;


%%%%%%%%% Translational element Tx differentials %%%%%%%
A(i,5)=-1;
A(i+1,5)=0;
```

```
A(i+2,5)=0;

%%%%%%%%% Translational element Ty differentials %%%%%%%
A(i,6)=0;
A(i+1,6)=-1;
A(i+2,6)=0;
%%%%%%%%% Translational element Tz differentials %%%%%%%
A(i,7)=0;
A(i+1,7)=0;
A(i+2,7)=-1;
 end

%%%% matrix of constants %%%%
for i=1:9;
    c=C(i);
    m=M(i);
    g=R(i);
    i=2*g-1;
L(i,1)=Xc(c)-(((Sc*((r11*X(m))+(r12*Y(m))+(r13*Z(m))))/1000)+Tx);
L(i+1,1)=Yc(c)-(((Sc*((r21*X(m))+(r22*Y(m))+(r23*Z(m))))/1000)+Ty);
L(i+2,1)=Zc(c)-(((Sc*((r31*X(m))+(r32*Y(m))+(r33*Z(m))))/1000)+Tz);
end

 N=A'*A; %%%%%Normal equation matrix %%%%%%%
Qxx=inv(N);%%%%%%%%% coffactor matrix %%%%%%
d=A'*-L;   %%%%% absolute vector %%%%%%%%
delta=Qxx*d; %%%%% corrections %%%%%%%%
%%updating the initial values %%%%%%
om=om+delta(1,1);
phi=phi+delta(2,1);
K=K+delta(3,1);
Sc=Sc+delta(4,1);
Tx=Tx+delta(5,1);
Ty=Ty+delta(6,1);
Tz=Tz+delta(7,1);

end
%%%%%%% coordinates of new points %%%%%
Par=[om;phi;K;Sc;Tx;Ty;Tz;]
for i=1:9;
    t=T(i);
    g=R(i);
    i=2*g-1;
XF(i,1)=(Sc*((r11*X(t))+(r12*Y(t))+(r13*Z(t)))/1000)+Tx;
XF(i+1,1)=(Sc*((r21*X(t))+(r22*Y(t))+(r23*Z(t)))/1000)+Ty;
XF(i+2,1)=(Sc*((r31*X(t))+(r32*Y(t))+(r33*Z(t)))/1000)+Tz;
end

Cord=XF;

  %%%%% Extracting coordinates %%%%%%
  for j=1;
      i=1:3:9;
      Cordx=Cord(i);
```

```
  i=2:3:9;
  Cordy=Cord(i);
  i=3:3:9;
  Cordz=Cord(i);
end

Coorninates=[Cordx Cordy Cordz]
```

```
Par =

   1.0e+05 *

   0.000001062439822
  -0.000000093138430
   0.000015373055892
  -0.006938398129888
   6.708963706946037
   2.232023296255391
   0.014329708109176


Coorninates =

   1.0e+05 *

   6.707501044372644   2.233589371303484   0.012823936947028
   6.702990725549419   2.235919704961493   0.013119235171436
   6.705089674299589   2.236024209664728   0.013018975318459
```

Published with MATLAB® R2014a